LA-UR-18-24210
Approved for public release; distribution is unlimited.

Title: MCNP6 UM Utility Programs

Author(s): Spencer, Joshua Bradly
Martz, Roger Lee
Alwin, Jennifer Louise

Intended for: Attila4MC Class

Issued: 2018-05-14

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

MCNP6 UM Utility Programs

Joshua B. Spencer, Roger L. Martz, Jennifer L. Alwin
XCP-3: Monte Carlo Codes, Methods and Applications

Abstract

The purpose of these slides is to introduce the user to MCNP6's utility programs for the unstructured mesh capability.

MCNP6 Unstructured Mesh Utility Programs

- **um_pre_op**
  - **Works with the Abaqus .abaq.inp to aid in problem setup.**
- **um_post_op**
  - **Perform various manipulations with the elemental edit output file (eeout).**
- **um_convert**
  - **Converts from .abaq.inp format to the mcnp internal .mcnpum format**

All three programs link with the Revised Eolus Grid Library (REGL) to use the same routines that MCNP6 uses when working with the unstructured mesh.

Please use versions of these utilities that are consistent with the version of MCNP6 in use.

MCNP6 Unstructured Mesh Utility Programs

- **Where to find these programs?**
  - **They come with the MCNP6 distribution.**
  - **Binaries should be in the “MCNP6/bin” directory.**
- **How to build these programs from the source?**
  - **Go to (cd) the MCNP6/Source directory and build MCNP6.**
  - **Go to (cd) the appropriate directory in MCNP6/Source/UM-Utilities.**
  - **Type: “make build”.**
  - **The binary will be built in that directory.**

MCNP6 Unstructured Mesh Utility Programs

- **All three programs operate from the command line.**
- **To see the help message for each program use the -h or --help input flags like:**

```
um_pre_op –h
```

or

```
um_post_op --help
```

MCNP6 Unstructured Mesh Utility Programs

Common command line options:

- Mutually exclusive.
- `-o, --output`
  - Program output is placed in a file with the argument name that follows on the command line.
- `-ex, --extension`
  - The output is placed in a file with a name built from the input file name followed by a '.' and the argument name immediately following on the command line.

MCNP6 UM Utility Programs

um_pre_op

The pre-processing program

Los Alamos NATIONAL LABORATORY EST.1943

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_pre_op --help

```
** PRE-PROCESSOR PROGRAM FOR UM CAPABILITY **

Functions:
1) Create MCNP input file from Abaqus .inp file
2) Convert MCNP simple lattice to Abaqus .inp file
3) Volume check the Abaqus .inp file and pseudo-cells
4) Element check the Abaqus .inp file

Command Line Arguments:
-b,   --back          background material for input file
-h,   --help          summary of features & arguments
-m,   --mcnp          generate MCNP skeleton input file --(1)
-o,   --output        output file name

-cf,  --controlfile  file with lattice conversion controls
-dc,  --datacards    data cards file to include
-ex,  --extension    output file extension
-ff,  --fillfile     file with lattice fill description
-lc,  --latconvert   convert simple lattice to Abaqus -- (2)
-vc,  --volcheck     volume check the .inp file      -- (3)
-ec,  --elementcheck element check the .inp file     -- (4)
-len, --length       scale factor for mesh dimensions
```

um_pre_op: Generating an MCNP input file

- **Purpose:** **Enables the user to get up and running quickly when there is an <u>existing</u> Abaqus .inp file.**
- **A skeleton file is generated. The degree of completeness rests with the completeness of the information in the data cards file.**
- **Additional CSG geometry must be added by hand.**
- **Advantage:** **creates the pseudo-cell cards and the matcell entry on the embed card using the logic that MCNP needs.**

um_pre_op : Generating an MCNP input file

- **To invoke:**

  `-m` **or** `--mcnp`

- **The .inp file can be anywhere on the line that is <u>not</u> input for another command line argument.**
- **Specify the background material with the `–b` argument.**

  `-b 7`

um_pre_op : Generating an MCNP input file

- Specifying the data cards file: `-dc`

  `-dc dc_cards`

- For each particle on the mode card, a default flux edit (`embee` card) is written to the input file.
- If active `imp` cards are present in the data cards file, they are written to the input file. Otherwise, default imp cards are written for each particle type.
- If an active `sdef` card is present, it is written to the file. Otherwise, a skeleton `sdef` card is written provided volume source elsets are present in the .inp file.

um_pre_op : Generating an MCNP input file

- **Example command line with the data cards argument and the –b argument to use material 7 from the .inp file as the background for the mesh universe.**

```
um_pre_op –m –o newinp abaqus.inp –dc cards –b 7
```


Los Alamos NATIONAL LABORATORY EST. 1943

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_pre_op: Converting a simple lattice geometry

- **What lattices are supported?**
  - **Simple lattice geometries that use the `fill` parameter along with the `lat` parameter (on a cell card).**
  - **Each voxel must have a homogeneous structure since each voxel is converted to a 1st order hexahedra with a homogeneous material assignment.**
  - **This is a restrictive subset of what MCNP supports.**
- **What is the result?**
  - **An Abaqus formatted .abaq.inp file that can be used with the `–m` or `--mcnp` option.**

um_pre_op : Converting a simple lattice geometry

- **Two input files are required:**
  - **The fill file specified with the `–ff` argument.**
  - **The control file specified with the `–cf` argument.**
- **The output file (i.e., the new .abaq.inp file) must be specified with the `–o` option; `-ex` is invalid here.**
- **In addition, a file named `lat2abq.summary` is created that contains information about the conversion process.**
  - **The information in this file can help the user adjust values for the hints keyword. (see below)**

um_pre_op : Converting a simple lattice geometry

The Fill File

- Contains only the fill information as it appears with the fill parameter on the MCNP lattice cell card.
- Example:

```
1   19R
2   7r   3   11R
2   2   4   2r
2   2   4   2r
3   4   3R   3   4   3R
```

um_pre_op : Converting a simple lattice geometry

The Control File

Example:

```
Jacksonville 1000 x 1000 x 31 model; 1 meter resolution
Deltas  100  100  100
fill    0:999  0:999  0:30
Origin  center
#
universe  1  -1.25000E-03  air
universe  2  -0.05         ext_building
universe  3  -0.01         int_building
universe  4  -1.2          ground
universe  5  -0.01         int_garage
universe  6  -0.087058     ext_garage
universe  7  -0.00125      air
#
exclude   1
extents   0 999  0 999  0 0
hints   200 200 50
threshhold  1
```

um_pre_op : Converting a simple lattice geometry

The Control File

- **The 1st line in the file is the title line.**
  - **Required as the 1st line**
  - **256 character limit**
  - **Inserted into the Abaqus .abaq.inp file**
- **Any line after the 1st line with either a #, %, or $ in the 1st column is treated as a comment line and ignored.**
- **All of the other parameters for this file are implemented with a set of keywords where the keyword appears at the beginning of the line before any values.**

um_pre_op : Converting a simple lattice geometry

The control file: deltas keyword

- Required
- 3 values that specify the length of the voxels in centimeters along the X, Y, and Z directions
  - Used to size the hexahedra
  - All hexahedra will have these dimensions

um_pre_op : Converting a simple lattice geometry

The control file: fill keyword

- Required
- 3 sets of values for the X, Y, and Z directions are needed in the same format that MCNP requires for this keyword on the lattice cell card.
- Each set consists of 2 lattice locations separated by a colon.
- The values specified for the fill keyword should be the full extents of the problem described in the fill file.
  - A subset of this geometry can be specified with the extents parameter.

um_pre_op : Converting a simple lattice geometry

The control file: universe keyword

- Required
- There may be as many universes specified on separate lines as needed to fully describe the problem.
- This concept of a universe is more restrictive than what MCNP allows in general.
- Each voxel should be homogenous so that 1 material can be assigned to it.
  - The universe numbers double as material numbers.

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_pre_op : Converting a simple lattice geometry

The control file: universe keyword (cont.)

- 3 values are required for each universe keyword.
- 1st: universe number
  - One for every universe number in the fill file.
  - Used as material numbers when describing the material elsets in the .abaq.inp file.
  - There is no default value; valid input is required.
- 2nd: material density (either number or physical)
- 3rd: universe / material name (128 characters max)
  - This name is used in creating material and part names.

um_pre_op : Converting a simple lattice geometry

The control file: exclude keyword

- Optional
- Contains a single universe number.
- Excludes the creation of any parts composed of this universe / material.
- Useful when the background material is an adequate substitute.
    - The MCNP calculation will be faster.

um_pre_op : Converting a simple lattice geometry

The control file: extents keyword

- Optional
- Used to select a contiguous subset from the fill range.
- Values are specified in the order:

  Lower X-index, upper X-index,
  Lower Y-index, upper Y-index,
  Lower Z-index, upper Z-index

um_pre_op : Converting a simple lattice geometry

The control file: hints keyword

- Optional, but highly recommended.
- 3 values, one for each direction, default is 9999999
- Specify number of columns (X), rows (Y), and planes (Z) into which the lattice should be segmented.
  - Serve as a guide to um_pre_op
- Parts are made from the segments.
  - All elements w/ the same material are lumped into a part whose name is derived from the `i,j,k` indices, the material number, and the material name.
- MCNP input processing for UM parts is quicker if parts don't have more than ~50,000 elements.

um_pre_op : Converting a simple lattice geometry

The control file: origin keyword

- Optional.
- Adjusts the location of the mesh origin.
- If not included, the origin defaults to 0, 0, 0.
  - i.e., absolute X, Y, Z location.
- If included, the mesh is shifted to specified value.
- With a value of CENTER, um_pre_op calculates the problem's center (using the deltas and the extents).

um_pre_op : Converting a simple lattice geometry

The control file: threshold keyword

- Optional.
- Contains a single integer value. Default is 1.
- um_pre_op makes a part when the number of elements in that part exceeds the threshold number.

um_pre_op : Converting a simple lattice geometry

- **Example command line to convert a simple lattice geometry:**

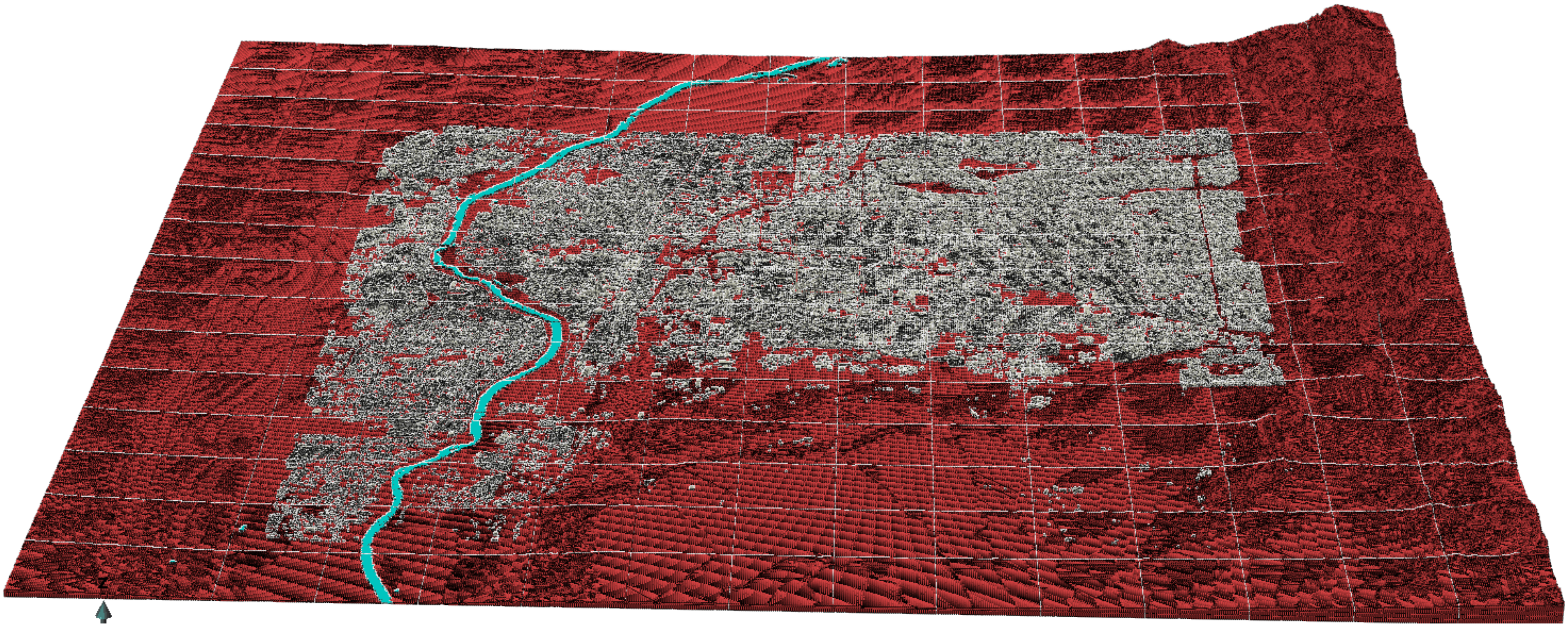
```
um_pre_op –lc –o lattice.inp –ff filefile –cf control
```


Los Alamos NATIONAL LABORATORY EST. 1943

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Albuquerque, NM

Hexahedral mesh converted from an MCNP lattice geometry



um_pre_op : Volume Checking

- **Enables the user to check certain volumes associated with the .abaq.inp file.**
  - **Check the finite elements' volumes against a specified value (see next slide)**
  - **Obtain volumes and masses for the pseudo-cells.**
    - **Masses -- provided densities are present in the .abaq.inp file.**
- **Results are written to the file specified with either the `–o` or `–ex` arguments.**

um_pre_op : Volume Checking

Finite element volumes

- **Any value after the –vc argument is treated as the test value.**
- **If value > 0, um_pre_op writes to the output file all elements and their volumes that are greater than or equal to the specified value.**
- **If value < 0, all elements and their volumes that are less than or equal to the value are written to the file.**
- **If value == 0, the test is for volumes less than or equal to 0.**

um_pre_op : Volume Checking

Example Volume Check Output File

```
simple warped cube
-
-   Data from file     : simple_cube_warped.inp
-   Created on         :  1-17-2014 @ 14: 0:58
-

    ---------------------------------------
    - Volume Check For Value  1.50000E+01 -
    ---------------------------------------

    Element       Volume
    -------    ------------
          1     7.81250E+00

    Elements with volumes <=  1.50000E+01 :        1


    ----------------------------------
    - Pseudo-Cell Volumes and Masses -
    ----------------------------------

    Cell    Instance    Part    Material    Denisty      Volume         Mass
    ----    --------    ----    --------    -------    -----------    -----------
       1           1       1           1   -8.95000    9.99219E+03    8.94301E+04
```


Los Alamos NATIONAL LABORATORY EST. 1943

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_pre_op : Volume Checking

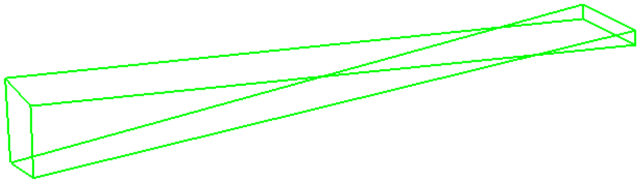
Example Volume Check Output File (cont.)

```
----------------------------------------------------------------------------
Instance    Name
--------    ----------------------------------------------------------------
       1    simple_cube-1


----------------------------------------------------------------------------
Material    Denisty    Name
--------    -------    -----------------------------------------------------
       1   -8.95000    material-copper_01
       2   -2.25000    material-graphite_02
```

um_pre_op : Element Checking

- Enables the user to check the .abaq.inp file for deformed and/or twisted elements.
  - Normal elements have a positive determinate of the Jacobian indicating that each point in the global space is mapped to an appropriate point in the master space.
- If a failed element is found (negative or zero Jacobian), the element number and location information are written to the terminal screen.
  - This same information as well as the results for the Jacobian evaluation at each Gauss and node point are written to the file specified with either the `–o` or `–ex` arguments.



um_pre_op : Element Checking

Example element check file

```
 test file for twisted 2nd order pents
-
-  Data from file    : um_cube_16pent2nd_twisted.inp
-  Created on        :  3-11-2014 @  8:31: 8
-

 --------------------------------
 - Checking Elements By Instance -
 --------------------------------

 Number  Name
 ------  ---------------------------------------------------------------------------
      1  part-cube-1

          Element:          2   failed.    Centroid:    1.50000E+00   1.22222E+00   6.66667E-01
                                              Nodes:         X             Y             Z
                                                   1    2.00000E+00   2.00000E+00   1.00000E+00
                                                   2    2.00000E+00   6.66667E-01   1.00000E+00
                                                   3    2.00000E+00   1.00000E+00   0.00000E+00
                                                   4    1.00000E+00   2.00000E+00   1.00000E+00
                                                   5    1.00000E+00   1.00000E+00   0.00000E+00
                                                   6    1.00000E+00   6.66667E-01   1.00000E+00
                                                   7    2.00000E+00   1.50000E+00   5.00000E-01
                                                   8    2.00000E+00   8.33333E-01   5.00000E-01
                                                   9    2.00000E+00   1.33333E+00   1.00000E+00
                                                  10    1.00000E+00   1.50000E+00   5.00000E-01
                                                  11    1.00000E+00   8.33333E-01   5.00000E-01
                                                  12    1.00000E+00   1.33333E+00   1.00000E+00
                                                  13    1.50000E+00   2.00000E+00   1.00000E+00
                                                  14    1.50000E+00   1.00000E+00   0.00000E+00
                                                  15    1.50000E+00   6.66667E-01   1.00000E+00
```

um_pre_op : Element Checking

Example element check file (cont.)

Determinate Values At Gauss Points

	Gauss Points			Jacobian
1	0.16667	0.16667	-0.57735	1.46E+00
2	0.66667	0.16667	-0.57735	4.10E-01
3	0.16667	0.66667	-0.57735	4.10E-01
4	0.16667	0.16667	0.57735	1.21E+00
5	0.66667	0.16667	0.57735	9.23E-01
6	0.16667	0.66667	0.57735	9.23E-01

Determinate Values At Master Space Nodes

	Gauss Points			Jacobian
1	0.00000	0.00000	-1.00000	2.00E+00
2	1.00000	0.00000	-1.00000	-6.67E-01
3	0.00000	1.00000	-1.00000	-6.67E-01
4	0.00000	0.00000	1.00000	6.67E-01
5	1.00000	0.00000	1.00000	6.67E-01
6	0.00000	1.00000	1.00000	6.67E-01
7	0.50000	0.00000	-1.00000	6.67E-01
8	0.50000	0.50000	-1.00000	-6.67E-01
9	0.00000	0.50000	-1.00000	6.67E-01
10	0.50000	0.00000	1.00000	6.67E-01
11	0.50000	0.50000	1.00000	6.67E-01
12	0.00000	0.50000	1.00000	6.67E-01
13	0.00000	0.00000	0.00000	2.00E+00
14	1.00000	0.00000	0.00000	6.67E-01
15	0.00000	1.00000	0.00000	6.67E-01

um_pre_op : Element Checking

Example element check file (cont.)

```
Element:          5   failed.   Centroid:   1.50000E+00   1.66667E+00   3.33333E-01
                                   Nodes:        X             Y              Z
                                       1   2.00000E+00   2.00000E+00   0.00000E+00
                                       2   2.00000E+00   1.00000E+00   0.00000E+00
                                       3   2.00000E+00   2.00000E+00   1.00000E+00
                                       4   1.00000E+00   2.00000E+00   0.00000E+00
                                       5   1.00000E+00   1.00000E+00   0.00000E+00
                                       6   1.00000E+00   2.00000E+00   1.00000E+00
                                       7   2.00000E+00   1.50000E+00   0.00000E+00
                                       8   2.00000E+00   1.50000E+00   5.00000E-01
                                       9   2.00000E+00   2.00000E+00   5.00000E-01
                                      10   1.00000E+00   1.50000E+00   0.00000E+00
                                      11   1.00000E+00   1.50000E+00   5.00000E-01
                                      12   1.00000E+00   2.00000E+00   5.00000E-01
                                      13   1.50000E+00   2.00000E+00   0.00000E+00
                                      14   1.50000E+00   2.00000E+00   1.00000E+00
                                      15   1.50000E+00   1.00000E+00   0.00000E+00

                                             Determinate Values At Gauss Points
                                                       Gauss Points          Jacobian
                                       1     0.16667    0.16667  -0.57735   -1.67E-01
                                       2     0.66667    0.16667  -0.57735   -1.67E-01
                                       3     0.16667    0.66667  -0.57735   -1.67E-01
                                       4     0.16667    0.16667   0.57735   -1.67E-01
                                       5     0.66667    0.16667   0.57735   -1.67E-01
                                       6     0.16667    0.66667   0.57735   -1.67E-01
```

um_pre_op : Element Checking

Example element check file (cont.)

Determinate Values At Gauss Points

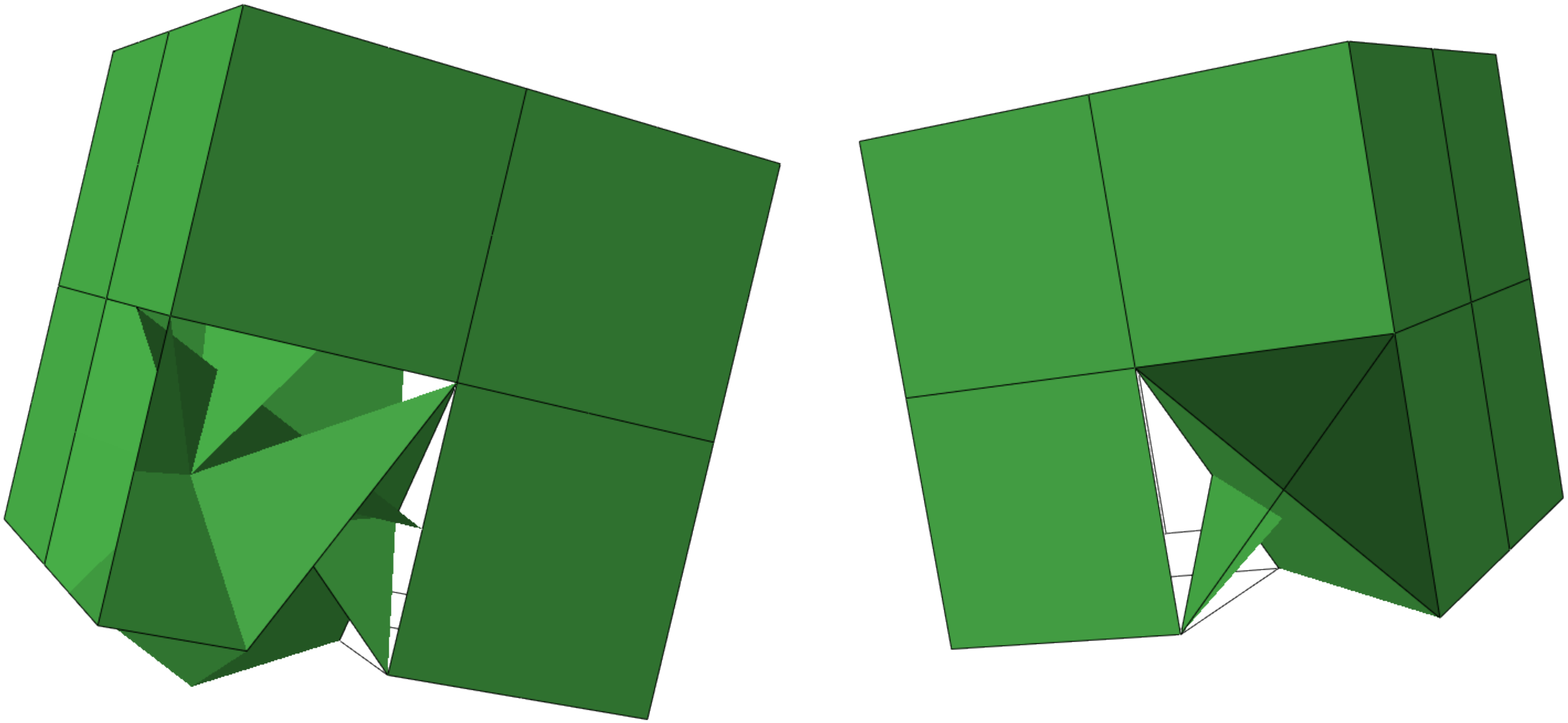
	Gauss Points			Jacobian
1	0.16667	0.16667	-0.57735	-1.67E-01
2	0.66667	0.16667	-0.57735	-1.67E-01
3	0.16667	0.66667	-0.57735	-1.67E-01
4	0.16667	0.16667	0.57735	-1.67E-01
5	0.66667	0.16667	0.57735	-1.67E-01
6	0.16667	0.66667	0.57735	-1.67E-01

Determinate Values At Master Space Nodes

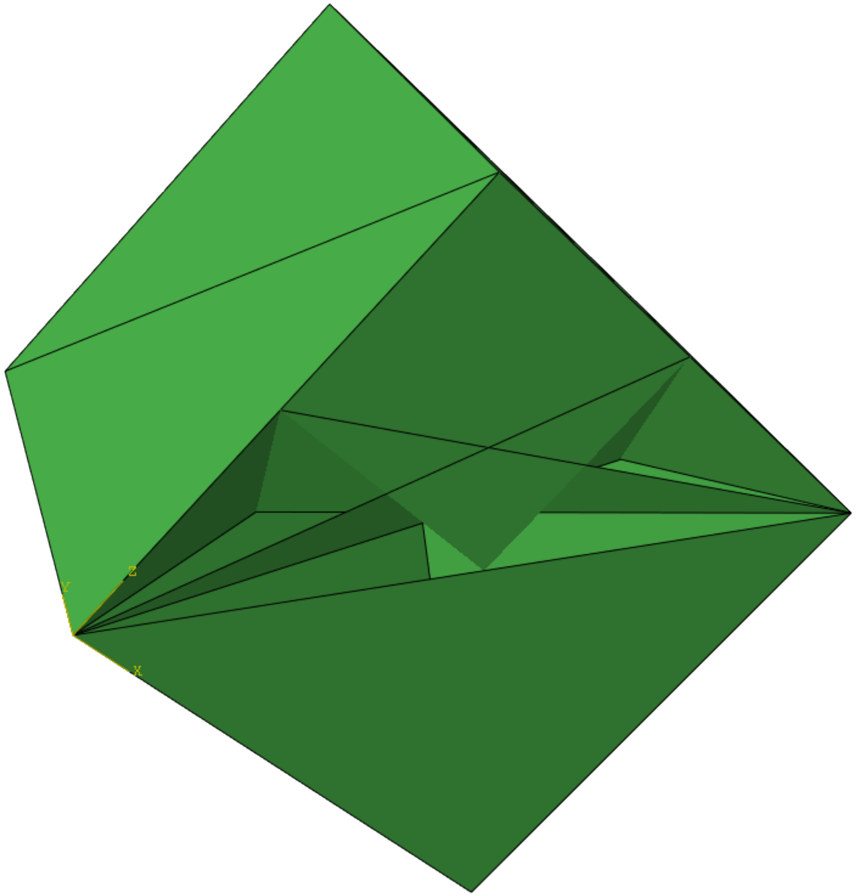
	Gauss Points			Jacobian
1	0.00000	0.00000	-1.00000	5.00E-01
2	1.00000	0.00000	-1.00000	5.00E-01
3	0.00000	1.00000	-1.00000	5.00E-01
4	0.00000	0.00000	1.00000	5.00E-01
5	1.00000	0.00000	1.00000	5.00E-01
6	0.00000	1.00000	1.00000	5.00E-01
7	0.50000	0.00000	-1.00000	5.00E-01
8	0.50000	0.50000	-1.00000	5.00E-01
9	0.00000	0.50000	-1.00000	5.00E-01
10	0.50000	0.00000	1.00000	5.00E-01
11	0.50000	0.50000	1.00000	5.00E-01
12	0.00000	0.50000	1.00000	5.00E-01
13	0.00000	0.00000	0.00000	-5.00E-01
14	1.00000	0.00000	0.00000	-5.00E-01
15	0.00000	1.00000	0.00000	-5.00E-01

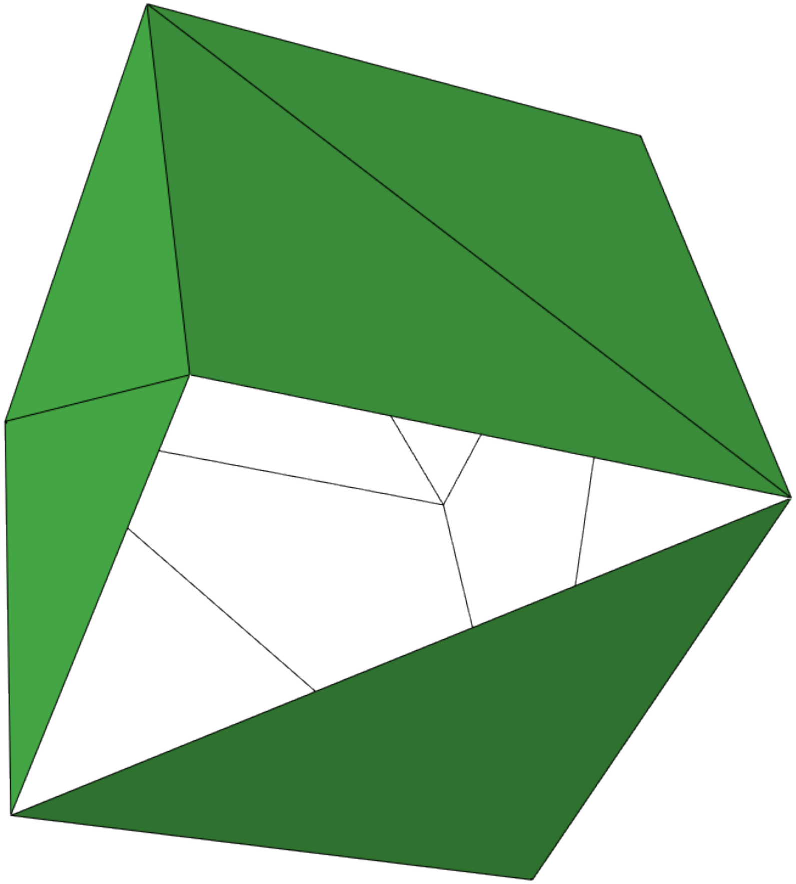
Total number of failed elements: 2

um_pre_op : Element Checking

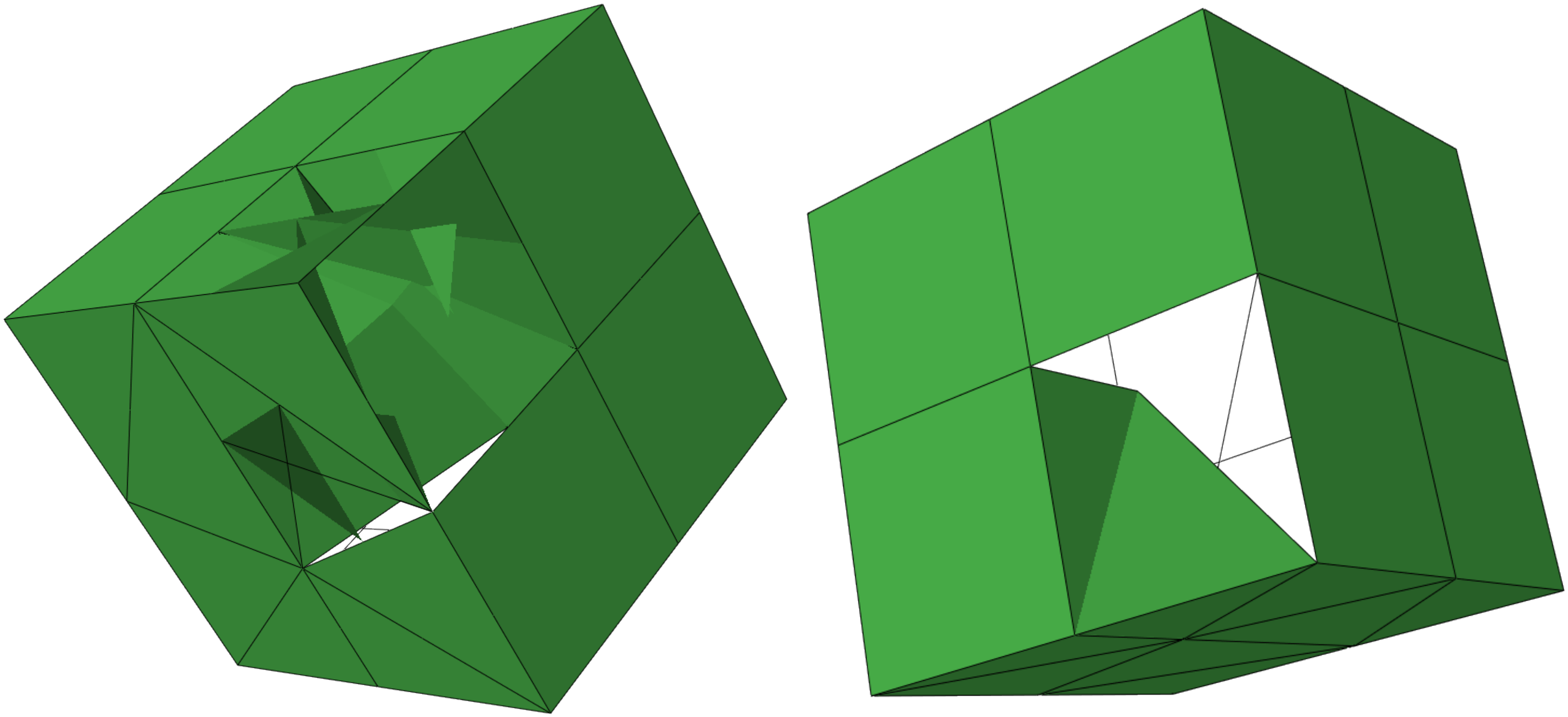


um_pre_op : Element Checking





um_pre_op : Element Checking



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

MCNP6 Utility Programs

um_post_op

The post-processing program

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_post_op --help

```
** UTILITY PROGRAM FOR UNSTRUCTURED MESH EEOUT FILE **

Functions:

1) add many eeout files into one
2) merge many eeout files into one
3) convert binary files into ascii files
4) generate vtk files for VisIt visualization
5) generate pseudo-tallies by instance
6) write a single edit to an ascii file
7) generate a histogram of edit errors

Command Line Arguments:

-h,   --help         summary of features & arguments

-a,   --add          add multiple files (no weighting)
-m,   --merge        merge multiple files
-o,   --output       single output file name
-p,   --pos          value range for wse and wsep
-bc,  --binconvert   convert binary file to ascii
-eh,  --errorhist    generate a histogram of edit errors
-ex,  --extension    multiple output file extension
-ta,  --tally        pseudo-tallies from file
-vtk, --vtkfile      generate ascii visualization file
-wse, --writesedit   write a single edit to file
```

um_post_op: Merging eeout file

- **The original intent for this utility program was to merge many eeout files into 1 file.**
  - **Files are expected to be from independent runs.**
  - **Results are weighted by the number of histories in each file.**
- **Does a consistency check on the header information in each file before attempting to merge.**
  - **Same number and type of elements, nodes, materials, edits, etc.**
  - **For all but the 1st file, a consistency check message is given.**
  - **Without consistency among the files, the program can NOT make a meaningful and successful merge.**

um_post_op: Merging eeout file

- **Files to be merged may be a mixture of ASCII and binary.**
  - **Output file is always ASCII.**
- **Example:**

```
um_post_op -m -o final eeout1 eeout2 ... eeoutN
```


Los Alamos NATIONAL LABORATORY EST. 1943
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_post_op: Adding eeout file

- **This capability adds (or collects) many eeout files into one.**
  - **Files are expected to be from different calculational runs on the same mesh geometry.**
    - **Still does consistency checks.**
  - **Results are not weighted by number of histories.**
  - **Already normalized results are simply added.**
  - **This capability is useful if there are different runs with different (independent) sources where it is acceptable to combine the results using superposition of scalar flux.**

um_post_op: Converting eeout files

- **Converts eeout files from binary format to ASCII.**
  - **Loss of precision since all double precision reals are written with only 6 significant digits.**
- **Any number of files may be specified on the command line.**
  - **Files are converted one at a time.**
  - **No consistency check required.**
  - `-ex` **argument required if more than 1 file specified for conversion.**
- **Example:**

```
um_post_op -bc -ex ascii eeout1 eeout2 ... eeoutN
```

um_post_op: Generating pseudo-tallies

- **A pseudo-tally for each instance from the corresponding edit in the eeout file is generated.**
- **The "tallies" are volume weighted:**

$$tally_i = \frac{\sum_{n=1}^{N} vol_n \cdot edit_n}{\sum_{n=1}^{N} vol_n}$$

$tally_i$ – **tally for instance 'i' from the corresponding edit**

vol_n – **volume of element 'n'**

$edit_n$ – **edit result of element 'n'**

N – **total number of elements in instance 'i'**

um_post_op: Generating pseudo-tallies

- **These results are termed "pseudo-tallies" since they are equivalent to an MCNP tally averaged over a volume (i.e., F4, F6, F7), but do NOT have an associated statistical uncertainty, tally fluctuation chart, etc.**
- **These pseudo-tallies are over instances and not pseudo-cells (unless there is a 1-to-1 correspondence).**
  - **There is no pseudo-cell information in the eeout file.**

um_post_op: Generating pseudo-tallies

- **Note the edit numbers in the following example:**

```
Pseudo-tallies for eeout file via um_post_op
Eeout file: eeout1007

Created on    :  4- 3-2012 @  9: 0:37

Prob ID       : simple cube, each element is a statistical set, 8 total
Calling Code  : MCNP6
Inp File      : inp1007
Outp File     : inp1007o
Runtpe File   : inp1007r
Geom Inp File : um1007.inp

NUMBER OF NODES    :                    27
NUMBER OF MATERIALS:                     1
NUMBER OF INSTANCES:                     1
NUMBER OF 1st TETS :                     0
NUMBER OF 1st PENTS:                     0
NUMBER OF 1st HEXS :                     8
NUMBER OF 2nd TETS :                     0
NUMBER OF 2nd PENTS:                     0
NUMBER OF 2nd HEXS :                     0
NUMBER OF COMPOSITS:                     1
NUMBER OF HISTORIES:                  1000
NUMBER OF REG EDITS:                    19
NUMBER OF COM EDITS:                     9

---------------------------------------------------------------------------
EDIT:     1 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_14

 Energy Bin Boundary:  1.00000E+36  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+33  Time Bin Multiplier  :  1.00000E+00

Instance  Name                                 Volume        Result
--------  ----                                 ------        ------
       1  simple_cube-1                   1.00000E+03   4.77743E-02
```

um_post_op: Generating pseudo-tallies

Psuedo-tally example (cont.)

```
 ---------------------------------------------------------------------------
 EDIT:     2 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_ENERGY_36

  Energy Bin Boundary:  2.00000E+00  Energy Bin Multiplier:  1.00000E+00
  Time Bin Boundary  :  1.00000E+00  Time Bin Multiplier  :  1.00000E+00

 Instance  Name                                   Volume        Result
 --------  ----                                   ------        ------
        1  simple_cube-1                     1.00000E+03   8.12612E-03

 ---------------------------------------------------------------------------
 EDIT:     3 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_2_ENERGY_36

  Energy Bin Boundary:  1.00000E+10  Energy Bin Multiplier:  1.00000E+00
  Time Bin Boundary  :  1.00000E+00  Time Bin Multiplier  :  1.00000E+00

 Instance  Name                                   Volume        Result
 --------  ----                                   ------        ------
        1  simple_cube-1                     1.00000E+03   7.54778E-03

 ---------------------------------------------------------------------------
 EDIT:     4 :: TALLY for  EDIT__PARTICLE_1__TIME_BIN_2_ENERGY_BIN_1_ENERGY_36

  Energy Bin Boundary:  2.00000E+00  Energy Bin Multiplier:  1.00000E+00
  Time Bin Boundary  :  1.00000E+39  Time Bin Multiplier  :  1.00000E+00

 Instance  Name                                   Volume        Result
 --------  ----                                   ------        ------
        1  simple_cube-1                     1.00000E+03   7.84947E-03

 ---------------------------------------------------------------------------
```

um_post_op: Writing a single edit

- Writes the edit results from a single edit in the eeout file to an ASCII file that is reformatted with detailed information (`–wse` or `–writesedit`).
- Information for each element:

element #	element type #	material #
density	volume	centroid location

- Results are ordered by increasing element number.

um_post_op: Writing a single edit

- **This request requires an edit # after the `–wse` or `–writesedit` argument.**
- **The correct edit number can be found in the output from the pseudo-tally option.**
- **Example:**

```
um_post_op –wse 1 –o eeout.wse eeout1
```

um_post_op: Writing a single edit

- **Filter the output using the `–p` or `–pos` arguments.**
  - **+1 or 1: only values > 0 are included in the edit.**
  - **-1: only values <= 0 are included in the edit.**
  - **If a real number is specified, its value is the decision point with the sign indicating whether the filter provides values greater than (+) or less than or equal to (-).**
- **Example:**

```
um_post_op -wse 1 -p -5.e-3 -o eeout.wse eeout1
```

um_post_op: Writing a single edit

WSE Example

```
-----------------------------------------------------------------
EDIT:    1 ::  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_14

 Energy Bin Boundary:  1.00000E+36  Energy Bin Multiplier:  1.00000E+00
 Time Bin Boundary  :  1.00000E+33  Time Bin Multiplier  :  1.00000E+00

-------------------------------------------------------------------------------------------------------------
Element   Type  Material    Density      Volume                  Centroid                       Result
                                                        X            Y            Z
--------  ----  --------  -----------  -----------  ------------  ------------  ------------  -----------
       1    6          1  1.87401E+01  1.25000E+02  -2.50000E+00  -2.50000E+00   7.50000E+00  4.50075E-02
       2    6          1  1.87401E+01  1.25000E+02  -2.50000E+00   2.50000E+00   7.50000E+00  4.71156E-02
       3    6          1  1.87401E+01  1.25000E+02  -2.50000E+00  -2.50000E+00   2.50000E+00  4.99385E-02
       4    6          1  1.87401E+01  1.25000E+02  -2.50000E+00   2.50000E+00   2.50000E+00  4.99248E-02
       5    6          1  1.87401E+01  1.25000E+02   2.50000E+00  -2.50000E+00   7.50000E+00  4.59879E-02
       6    6          1  1.87401E+01  1.25000E+02   2.50000E+00   2.50000E+00   7.50000E+00  5.14196E-02
       7    6          1  1.87401E+01  1.25000E+02   2.50000E+00  -2.50000E+00   2.50000E+00  4.33516E-02
       8    6          1  1.87401E+01  1.25000E+02   2.50000E+00   2.50000E+00   2.50000E+00  4.94486E-02
```

um_post_op: Writing a single edit by postion

- **Similar to `–wse` except that the output is ordered by increasing position (i.e., x, y, z location).**
- **The appropriate arguments to use on the command line are:**

  `–wsep` or `–writeseditpos`

- **Value filtering, as previously described, works here.**

um_post_op: Generating an error histogram

- Write error histograms to an output file for all of the edits in the eeout file for which errors were requested.
- The number of histogram bins is specified directly after the **–eh** argument. Default = 10.
- The error bins are defined such that the smallest error is assigned to the 1st bin and the largest error is assigned to the last bin.
  - Bins are evenly spaced between the 1st and last bins.
- Example:

```
um_post_op -eh 20 -o my_error_histogram eeout1
```

um_post_op: Generating an error histogram

Example Error Histogram File

```
Write error histograms for eeout file via um_post_op
Eeout file: block01_6part_6type.eeout

Created on    :  9- 5-2012 @ 13:14:34

Prob ID       : block01  8x8x6  6 parts, 6 element types
Calling Code  : MCNP6
Inp File      : block01mgv1
Outp File     : outq
Runtpe File   : runtpf
Geom Inp File : job_block_6part_6type_01.inp

NUMBER OF NODES    :                     1258
NUMBER OF MATERIALS:                        6
NUMBER OF INSTANCES:                        6
NUMBER OF 1st TETS :                       30
NUMBER OF 1st PENTS:                        8
NUMBER OF 1st HEXS :                      128
NUMBER OF 2nd TETS :                       29
NUMBER OF 2nd PENTS:                        8
NUMBER OF 2nd HEXS :                      128
NUMBER OF COMPOSITS:                        0
NUMBER OF HISTORIES:                  1000000
NUMBER OF REG EDITS:                        2
NUMBER OF COM EDITS:                        0
```

um_post_op: Generating an error histogram

Example Error Histogram File (cont.)

```
---------------------------------------------------------------
 EDIT:  EDIT__PARTICLE_1__TIME_BIN_1_ENERGY_BIN_1_FLUX_4

     Energy Bin Boundary:  1.00000E+10  Energy Bin Multiplier:  1.00000E+00
     Time Bin Boundary  :  1.00000E+39  Time Bin Multiplier  :  1.00000E+00
     ----------------------------------------------------
     Results for Instance #     1  :: part-end_quad_hex-1

     Minmum Error        :  1.64393E-02
     Maximum Error       :  1.70379E-02
     Bin Width           :  2.99308E-05

     ------   --------   --------   --------    ----------
     Bin       Upper     Absolute   Relative    Cumulative
     Number    Bound      Number      (%)          (%)
     ------   --------   --------   --------    ----------
        1   1.6469E-02          1     0.7812        0.7812
        2   1.6499E-02          1     0.7812        1.5625
        3   1.6529E-02          3     2.3438        3.9062
        4   1.6559E-02          5     3.9062        7.8125
        5   1.6589E-02          0     0.0000        7.8125
        6   1.6619E-02          7     5.4688       13.2812
        7   1.6649E-02          6     4.6875       17.9688
        8   1.6679E-02         14    10.9375       28.9062
        9   1.6709E-02          5     3.9062       32.8125
       10   1.6739E-02          6     4.6875       37.5000
       11   1.6769E-02         13    10.1562       47.6562
       12   1.6798E-02         14    10.9375       58.5938
       13   1.6828E-02         12     9.3750       67.9688
       14   1.6858E-02         11     8.5938       76.5625
       15   1.6888E-02          5     3.9062       80.4688
       16   1.6918E-02         10     7.8125       88.2812
       17   1.6948E-02          4     3.1250       91.4062
       18   1.6978E-02          7     5.4688       96.8750
       19   1.7008E-02          3     2.3438       99.2188
       20   1.7038E-02          1     0.7812      100.0000
```

um_post_op: Generating an error histogram

Example Error Histogram File (cont.)

```
(Results for instances 2 through 6 were removed to make this example shorter.)

     ---------------------
     Results Over All Mesh

     Minmum Error       :  9.33224E-03
     Maximum Error      :  1.95299E-02
     Bin Width          :  5.09881E-04

     ------  --------  --------  --------   ----------
     Bin      Upper    Absolute  Relative   Cumulative
     Number   Bound     Number     (%)         (%)
     ------  --------  --------  --------   ----------
        1   9.8421E-03        4    1.2085       1.2085
        2   1.0352E-02        8    2.4169       3.6254
        3   1.0862E-02        0    0.0000       3.6254
        4   1.1372E-02        0    0.0000       3.6254
        5   1.1882E-02        4    1.2085       4.8338
        6   1.2392E-02        1    0.3021       5.1360
        7   1.2901E-02        0    0.0000       5.1360
        8   1.3411E-02        3    0.9063       6.0423
        9   1.3921E-02        3    0.9063       6.9486
       10   1.4431E-02        9    2.7190       9.6677
       11   1.4941E-02        9    2.7190      12.3867
       12   1.5451E-02        4    1.2085      13.5952
       13   1.5961E-02        0    0.0000      13.5952
       14   1.6471E-02       15    4.5317      18.1269
       15   1.6980E-02      241   72.8097      90.9366
       16   1.7490E-02       18    5.4381      96.3746
       17   1.8000E-02        5    1.5106      97.8852
       18   1.8510E-02        6    1.8127      99.6979
       19   1.9020E-02        0    0.0000      99.6979
       20   1.9530E-02        1    0.3021     100.0000
```

MCNP6 Utility Programs

um_convert

Convert from .abaq.inp format to .mcnpum

Los Alamos NATIONAL LABORATORY EST. 1943

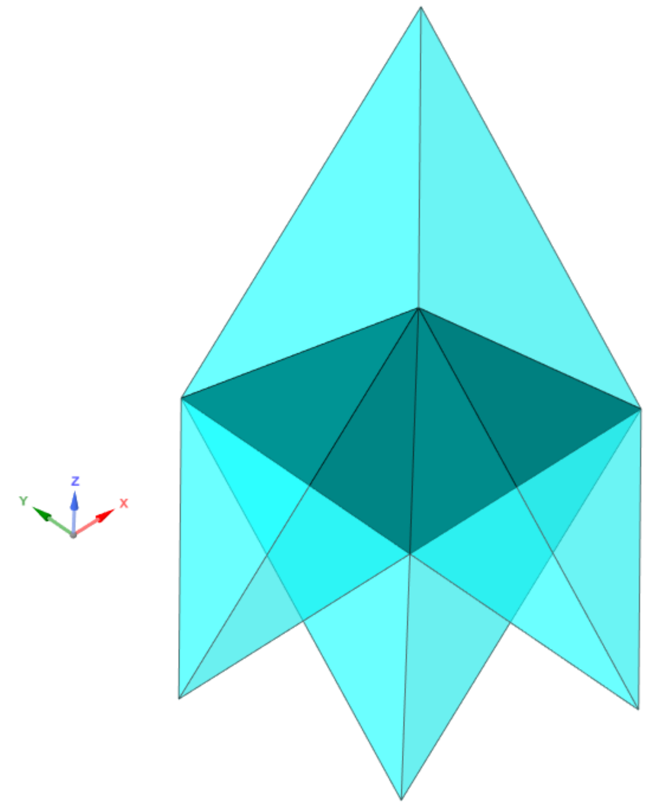
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

um_convert - h

- 1) Convert ABAQUS inp file to mcnpum file
- Command Line Arguments:
  - -h, --help summary of features & arguments
  - -b, --binary create mcnpum in binary format
  - -a, --abaqus ABAQUS input file -- (1)
  - -l, --length length conversion factor
  - -o, --output um_convert output file name
  - -t, --threads number of threads
  - -um, --mcnpum mcnpum output file name

Advantage of Using um_convert

- **Fast Read in performance in mcnp, especially when in binary format**
- **MCNP will convert to geometry in .abaq.inp files to .mcnpum for improved internal tracking performance every time it runs if the user does not supply the pre-converted file**
- **.mcnpum format includes additional data such as the nearest neighbors for each element in the um**



- Element containing tracked Particle
- Nearest neighbors to current element

um_convert Length Conversion Factor

- **The length conversion factor in the um_convert utility serves the same function as the length parameter in the embed card in the mcnp input file, that is to convert the native length units in the .abaq.inp file to centimeters**
  - **example if the .abaq.inp file is in mm then the length parameter should be set to 0.1**
  - **Abaqus meshes generated in Attila4MC should always come in centimeters so this parameter shouldn't need to be specified**
- **If a user supplies a pre-converted mesh geometry to mcnp, it is assumed that the length conversion has already been made so any length parameter in the mcnp embed card is ignored**